Verbio Software Reference

Grammar Manager User's Manual

Verbio Technologies, S.L.

Verbio Software Reference: Grammar Manager User's Manual

Verbio Technologies, S.L.

Published September, 2011
Copyright © 2011 Verbio Technologies, S.L.

Table of Contents

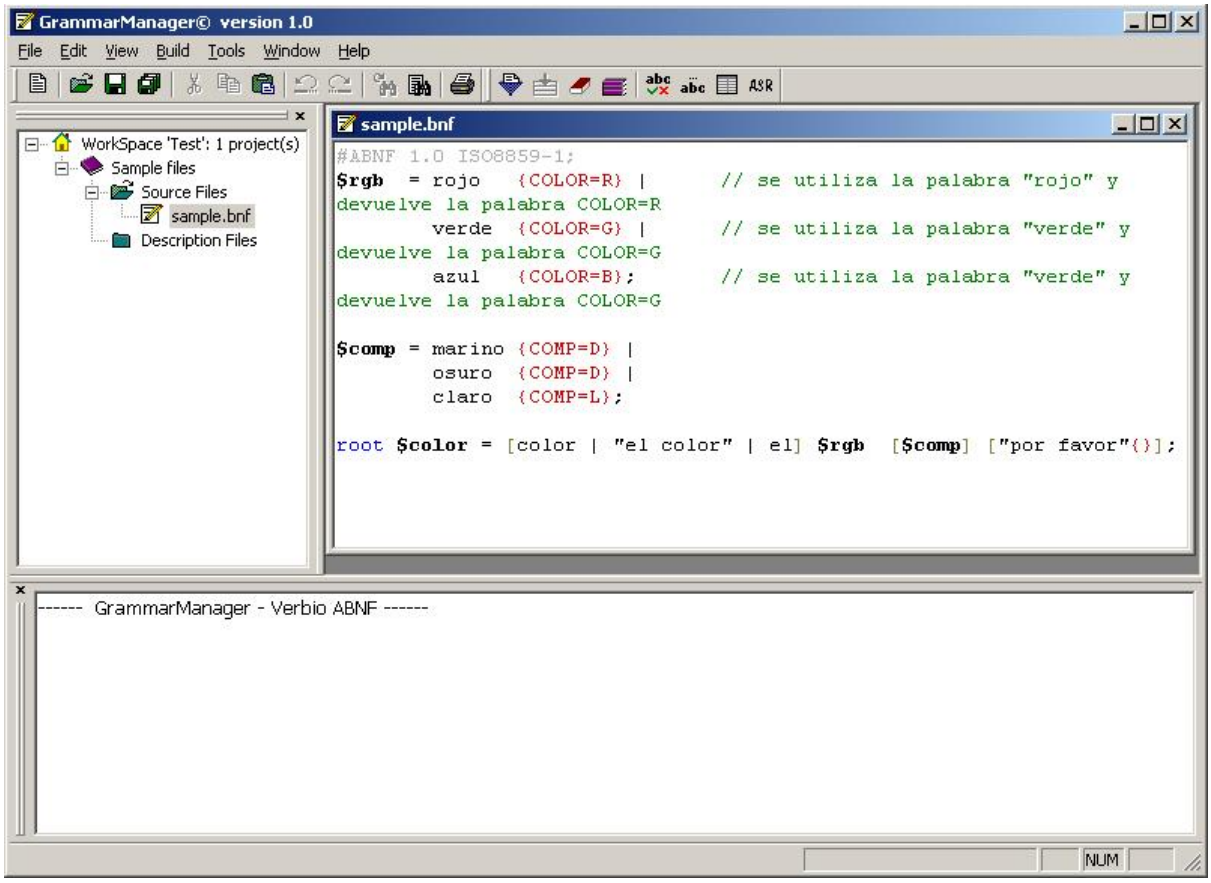
1. User Interface 1
    1. General description 1
    2. Available Menus 1
    3. Windows and bars 2
    4. Toolbars 2
        4.1. The Main toolbar 2
        4.2. The Build toolbar 3
2. The WPF architecture 5
    1. What is the WPF architecture? 5
    2. Why do I need a WorkSpace? 5
    3. Why do I need a WorkSpace? 5
    4. How are the files classified? 6
    5. Thanks, but I prefer don't work with the WPF architecture 6
3. Compiling and Building 7
    1. The Compilation process 7
    2. Building, cleaning and rebuilding all 8
    3. Running, testing and using grammars 8
        3.1. Running grammars 9
        3.2. Testing grammars 9
        3.3. Trying grammars 9
4. Grammar Manager Tutorial 11
    1. Create a New WorkSpace and Project 11
        1.1. Creating a workspace and a project in one step 11
        1.2. Creating a workspace and a project in two steps 11
    2. Add files to the folders 11
    3. Test your grammar 12
Index 13

Chapter 1. User Interface

1. General description

Grammar Manager is a WIN-32 Multithread application. Its visual environment makes this application an easy and powerful tool for developing, managing and testing Augmented BNF (ABNF) grammars for the *Verbio* Automatic Speech Recognition (ASR) engine.

Grammar Manager is built on the WPF (Workspace-Project-File) architecture which helps you to organize better your work. It offers you compilation and testing tools, to check for syntax and phonetic transcription errors, and lets you test the performance of your grammars. As most Windows applications, Grammar Manager it's made up of menus, toolbars and floating dockable bars. All them are described in the next sections.


GrammarManager© version 1.0
File Edit View Build Tools Window Help
WorkSpace 'Test': 1 project(s)
Sample files
Source Files
sample.bnf
Description Files
sample.bnf
#ABNF 1.0 ISO8859-1;
$rgb = rojo {COLOR=R} | // se utiliza la palabra "rojo" y devuelve la palabra COLOR=R
verde {COLOR=G} | // se utiliza la palabra "verde" y devuelve la palabra COLOR=G
azul {COLOR=B}; // se utiliza la palabra "verde" y devuelve la palabra COLOR=G
$comp = marino {COMP=D} |
osuro {COMP=D} |
claro {COMP=L};
root $color = [color | "el color" | el] $rgb [$comp] ["por favor"{}];
------ GrammarManager - Verbio ABNF ------
NUM

Main window, menus, toolbars and floating windows in the *Verbio Grammar Manager* environment.

2. Available Menus

Grammar Manager has only one menu made up of 7 submenus: File, Edit, View, Build, Tools, Window and Help. Some of the functions that you can find in these submenus are also accessible from the toolbars.

- The *File* submenu allows you to load, save and create new documents, projects or workspaces. It also includes a useful MRW (Most Recent WorkSpaces) list and a MRF (Most Recent Files) list, that let you find and open your lasts grammars and workspaces.
- The *Edit* submenu has the common edit functions that all the editors have. That includes the edit, copy and paste functions, the undo and redo capabilities and the find, find next and replace options.

- The *View* submenu allows you to show or hide `Grammar Manager` toolbars, docking bars and its status bar. It also lets you switch the drawing mode of the menus between the `XP` drawing mode and the classic `Windows` 9x and 2k drawing mode.
- The *Build* submenu allows you to compile, run and test grammar files, as well as build and clean projects. All these functions are described in detail in further sections.
- The *Tools* submenu lets you choose some options that affect to the grammar compilation, testing and running.
- The *Window* submenu lets you work with several windows at a time, selecting their position in the main frame.
- Finally, the *Help* submenu lets you check the documentation of Grammar Manager and know more about its authors.

3. Windows and bars

`Grammar Manager` is a MultiDocument Application. This means that you can have more than one document opened at the same time. To switch between documents press Ctrl+Tab. You can drag a bar and dock it to another part of the frame by clicking on the draggable zone and moving it. Double-clicking on the draggable zone of the bars will switch its style between docking and floating.

- The *WorkSpace* bar displays the workspace hierarchy (projects, folders and files). Its items have a context menu that pops-up with the right-click of the mouse.
- The *Output* bar displays the output messages generated during the compilation and building processes.
- The *Run* bar displays the generated samples for a compiled grammar.

4. Toolbars

`Grammar Manager` has two toolbars.

- The *Main* toolbar includes the main open/save options and the editing options.
- The *Build* toolbar includes the compile and build options.

Both of them can be hided/showed by selecting the `View - Toolbars` options from the main menu. Next sections give you a quick description about each button of these toolbars.

4.1. The Main toolbar

The Main Toolbar

By clicking on the buttons of the main toolbar you will (left to right):

- Open a new document, project or workspace. The `new dialog` will pop-up.
- Open an existing document, project or workspace. The `open dialog` will pop-up.
- Save the active document.
- Save all the opened documents as well as the workspace and its projects.

- Remove selected data from the document and stores it on the clipboard.
- Copy the selection to the clipboard.
- Insert the contents of the clipboard at the insertion point.
- Undo the last editing action.
- Restore the last undo.
- Repeat the last find.
- Find a word or expression in the current document. The `find dialog` will pop-up.
- Print the active document. The `print dialog` will pop-up.

4.2. The Build toolbar

The Build Toolbar

By clicking on the buttons of the build toolbar you will (left to right):

- Compile the current document.
- Build the active project.
- Clean the active project.
- Clean and build all the projects in the workspace.
- Test the active grammar file.
- Use future.
- Run the active grammar file.
- Recognize with the active grammar file.

Chapter 2. The WPF architecture

1. What is the WPF architecture?

The `WPF (WorkSpace-Project-Files)` architecture is just a way to organize your work in a pyramidal hierarchy starting from the base with grammar files and ending at the top with the workspace. Between them you will find projects and folders. The complete hierarchy is as follows: Only one workspace can be opened at the same time. The workspace contains projects. Each workspace can have as many projects as you want. Each project has three folders: Source Files, Description Files and Setting Files. Each folder contains as many files as you want.

Organize your work with the `WPF` (WorkSpace-Project-Files) architecture, may seem useless at the beginning, but it turns extremely powerful when you get used to it.

2. Why do I need a WorkSpace?

Probably, the first question that comes to your mind is why you need a WorkSpace. The answer is easy: "To detach work environments".

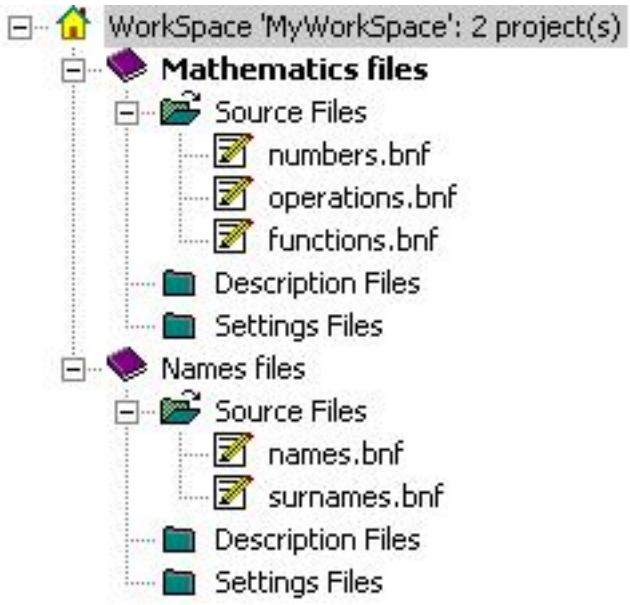
Imagine that you are a web server, and you are maintaining two `VoiceXML` voice portals with different subjects: one of them is a sport voice portal and the other one a science voice portal. Each voice portal has its own grammars. When you want to work with the grammars of the sport voice portal you would probably like to have all them together. Won't you? Well, this is basically the purpose of the workspace. In this case, you would have 2 workspaces (one per voice portal) and when you want to work with one voice portal, you would open it and work with only its grammars and their settings. The workspaces are made up of projects. The next question is then obvious. Why do you need Projects?

3. Why do I need a WorkSpace?

Once you have understood that it's important to detach different works environments that have different purposes and probably different grammar files, think how useful would be to have a way for classifying files into your workspace. Let's say that you create a workspace which contains all the grammars that you have in a generic voice portal. Imagine that these grammars deal with mathematic expressions, names and surnames:

```
numbers.bnf, functions.bnf, operations.bnf, names.bnf and surnames.bnf
```

Having them arranged into two groups would be nice to detach concepts. Isn't it? One group would contain the mathematic expressions and the other one the names and surnames. This is what the projects are for. In this case you could have one project called `Mathematics` and another called `Names`. All in one workspace called `MyWorkSpace` or `MyVoicePortal`.



`Mathematics` and `Names` projects in `MyWorkSpace` workspace.

Furthermore, having files classified into projects, makes easier to share grammars between users. Imagine that somebody asks you for grammars with mathematic stuff. You would just tell him to add your `Mathematics` project into his workspace...that easy!!

4. How are the files classified?

The files are classified into 3 different folders:

- The `Source Files` folder contains the grammar files which are written following the `ABNF` standard.
- The `Description Files` folder contains the description files which are used to describe and comment the grammars. They are optional.
- The `Settings Files` folder contains the settings files which are used to specify grammar properties such as n-best or language. They are not used in the current Grammar Manager version.

5. Thanks, but I prefer don't work with the WPF architecture

We strongly recommend to use the WPF architecture, but if for any reason you prefer not to work with it, you can still open, close, edit, compile, test and run grammar files. The procedures are the same as for the case you are working within a workspace.

Chapter 3. Compiling and Building

1. The Compilation process

The compilation process checks for syntax errors as well as for errors occurred during the phonetic transcription of your grammar. You can select whether to generate *Verbio* files (*.trc) or don't. If you do it, syntax errors and phonetic transcription errors will be checked. If you don't, only the syntax errors will be checked. To generate *Verbio* files, select the `Tools - Grammar Settings` option from the main menu. The `Grammar Settings` dialog will pop up. Select the `Generate Verbio output files` options and choose the IP address of the PC where *Verbio VoxServer* is installed (127.0.0.1 for local server or non client-server architecture).

Grammar settings dialog

Caution

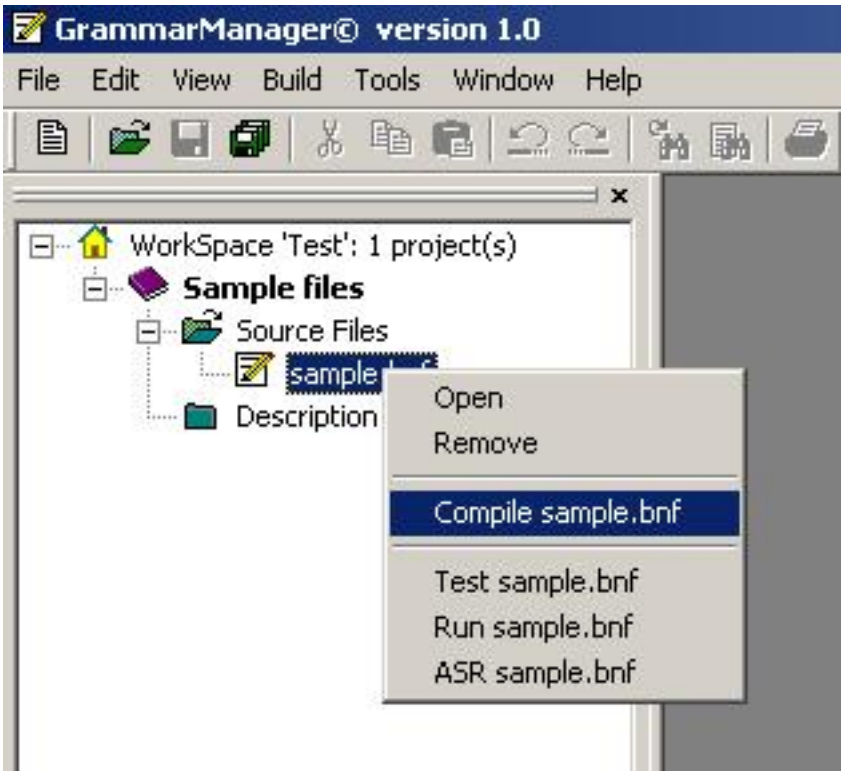
To generate *Verbio* files you need to have at least one `Verbio ASR engine` accessible from your PC.

If a syntax error is detected during the compilation process, the output bar will notice you that the *.net or *.trx/*.grm files have errors and the line error will be highlighted in the document file. If a phonetic transcription error is detected, the output bar will notice you that the *.trc file has errors and the word that has errors in its phonetic transcription will be highlighted in the document file.

Note

Be sure you have selected the correct language configuration for the given grammar.

To `Compile` a file, open it and click on the compile button or select the `Build - Compile` option from the main menu. Only the current active document file will be compiled. To compile a file regardless of whether it has been opened or not, right click on the file name in the workspace view and select the `Compile` option from the context menu.



Compile a grammar file using the context menu.

2. Building, cleaning and rebuilding all

Build, rebuild and clean options are in the context of projects and workspaces.

Build a project means to compile all the files within that project. If you click on the build button,the active project will be built. The active project is the project which name appears in bold type in the workspace view. You can set a project as the active project by right clicking on the project and selecting the `Set as Active Project` option from the context menu. You can also build the active project by selecting the `Build - Build` option from the main menu. To build a project regardless of whether it is the active project or not, right click on the project name in the workspace view and select the `Build (selection only)` option from the context menu. The build option automatically skips the files that have been already compiled successfully in a previous compilation process and have not been modified. If the build process finds errors during the compilation, the output bar will alert the user and at the end of the building process will display the number of errors found.

Clean a project means to delete the output files generated during the compilation process (*.net, *.trx/*.grm and *.trc) for the files contained in the project. This will force the next compilation to regenerate them and so the next build will not skip any files. The clean button cleans only the active project files. You can also clean the active project by selecting the `Build - Clean` option from the main menu. To clean a project regardless of whether it is selected as the active project or not, right click on the project and select the `Clean (selection only)` option from the context menu.

The *Rebuild All* option will clean all the projects in the workspace and rebuild them. To `Rebuild All` click on the rebuild all button or select the `Build - Rebuild All` option from the main menu.

3. Running, testing and using grammars

`Grammar Manager` gives you the choice to test the grammars you have just written. After compilig them without errors, three tools are available to test its performance: the run, the test and the try tool. All of them need the *Verbio* file (*.trc) of your grammar. To generate it, select the `Generate Verbio output files` option in the `Grammar Settings` dialog (`Tools - Grammar Settings` option from the main menu). Before running, testing and trying grammars, `Grammar Manager` will check if your document has been modified after last compilation process. If it does, it will ask you to compile it again before starting the process.

3.1. Running grammars

To run the current grammar file click on the Run button, press Ctrl+F5 or select the `Build - Run` option from the main menu. The run bar will appear (if it was not already visible), and display some random generated samples according to the grammar rules. The number of generated samples as well as the grammar language (used for the phonetic transcription) can be changed through the `Grammar settings` Dialog (`Tools - Grammar Settings` from main menu).

3.2. Testing grammars

To test the current grammar file click on the Test button, press F5 or select the `Build - Test` option from the main menu. The `Test Dialog` will pop-up. Type the text you want to check whether is a possible word construction according to the grammar rules or it is not, and click on the `Check` button. An icon will appear validating or invalidating the word construction. For example, using a boolean grammar:

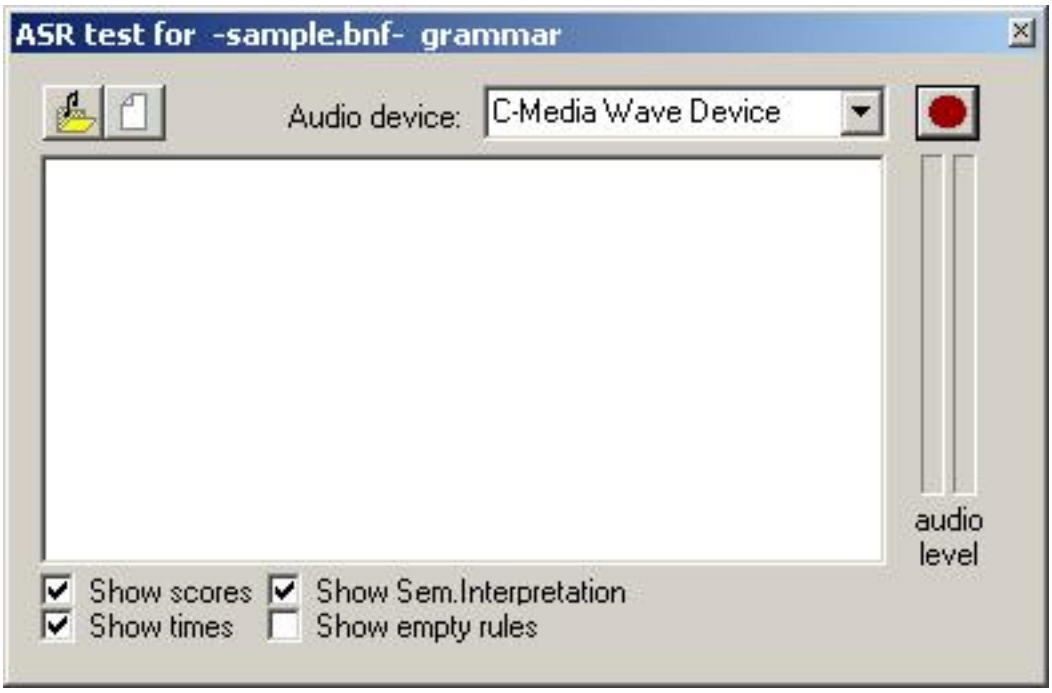
```
#ABNF 1.0 ISO8859-1;

root $boolean = $command;

$command =
        true |
        false
;
```

Successful sentence.

Incorrect sentence.

3.3. Trying grammars

If a sound car is available on the computer, it's possible to try the grammar in a real environment: the voice world. To try the current grammar file click on the ASR button or select the `Build - ASR Test` option from the main menu. The `ASR Test` will pop-up.



The ASR Test pop-up.

To try the grammar contents, press the Record button and speak normally. To stop the recognition proccess, press de Stop button or remain in silence. When finished, the recognition results will appear showing the selected parameters:

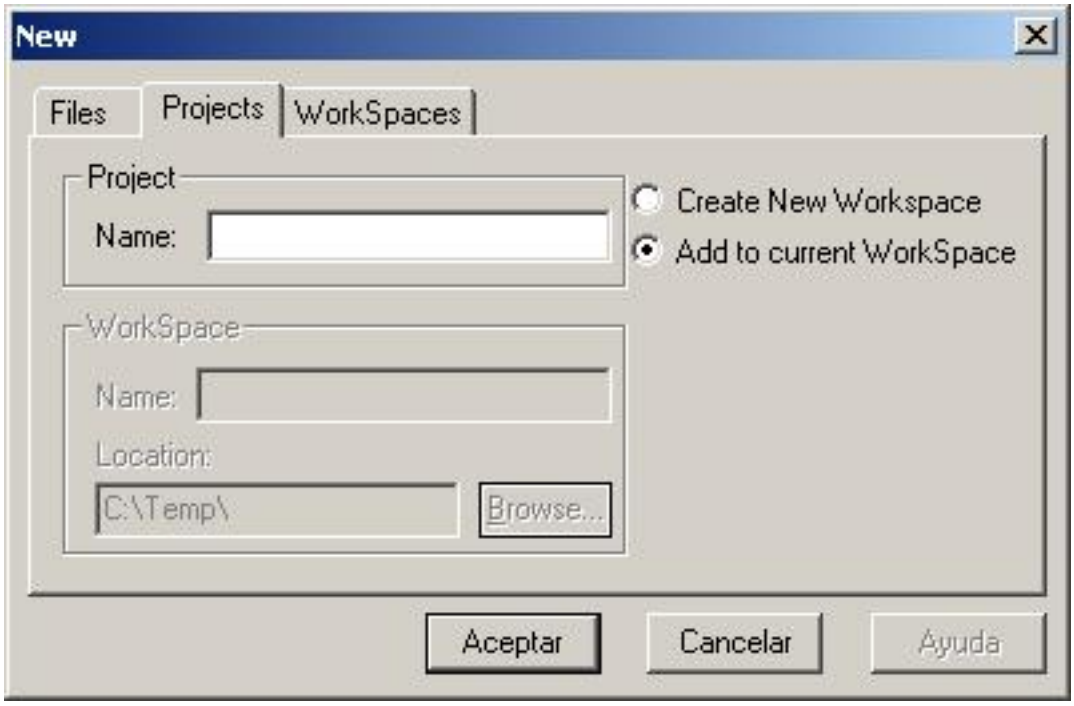
- *Scores*: Measure of the confidence.
- *Times*: Initial time and final time (in seconds) for every utterance.
- *Semantics interpretation*: Spoken words.

Chapter 4. Grammar Manager Tutorial

This tutorial is a practical guide that enumerates the steps to follow to get a workspace with a project and files inside.

1. Create a New WorkSpace and Project

Select the `File - New` option from the main menu or click on the new button from the main toolbar to open the *New* dialog.



The New - Projects Dialog.

This dialog allows you to generate Files, Projects or Workspaces. The hierarchy between them is the following:

A workspace contains projects. Only one workspace can be opened simultaneously. Loading or creating a new workspace will close the current one.

A project contains files. These files are classified into three folders : Source Files, Description Files and Setting Files.

1.1. Creating a workspace and a project in one step

Select the `projects` tab. Type a name for the workspace and a name for the project. `Grammar Manager` will select a default location for them. We strongly recommend to keep this configuration, but is up to the user to change it.

1.2. Creating a workspace and a project in two steps

The same operation we just did in the previous section, can be done in two steps: First, generate a blank workspace (with no projects inside), by selecting the `WorkSpaces` tab in the `New` dialog. Give it a name and a location. When this is done, open again the `New` dialog and select the projects tab with the `Add to current workspace` option selected. Give a name to the project and then click OK.

2. Add files to the folders

You can either load or create files and add them to the folders.

To *Load* files, right click on the folder you want to add the files and select the `Add Files to Folder...` option. The `Load` Dialog will pop-up. Select the files you want to add to the folder and press Ok. Multiple-files selection is allowed.

To *Create* new files, select the `Add New Files to Folder` option. The `New` dialog will pop-up with the `Files` tab selected. Choose the name, the location and the type of file you want to create and press OK.

3. Test your grammar

Once you have grammar files, you can compile, test or execute them.

To *Compile* a file, open it and click on the Compile button. Only the current active document will be compiled. You can also compile a file without opening it, by right clicking on the file in the workspace view and selecting the compile option. The `Output Bar` will notice you about errors during the compilation process. If no errors ocurred, you can run and test the grammar.

To *Run* the grammar, open it and click on the Run button. `Grammar Manager` will display generated sentences that obey to the current active grammar. You can also run a grammar without opening it, by right clicking on the grammar file in the workspace view and selecting the run option.

To *Test* the grammar, open it and click on the Test button. The current active grammar document will be tested. You can also test a grammar without opening, by right clicking on the grammar file in the workspace view and selecting the test option.

Index

B

Build
  building, 8

C

Compile
  compiling, 7

D

Description, 1

F

File classification, 6

G

Grammar processing, 7

M

Menu, 1

P

Projects, 5

R

Run
  running, 8

T

Toolbars, 2
  Build toolbar, 3
  Main toolbar, 2
Tutorial
  adding files, 11
  creating workspace, 11
  testing files, 12

W

Windows, 2
Workspaces, 5
WPF, Workspace-Project-Files architecture, 5